Vision and Challenges for Multicore Computing in Future Space Missions

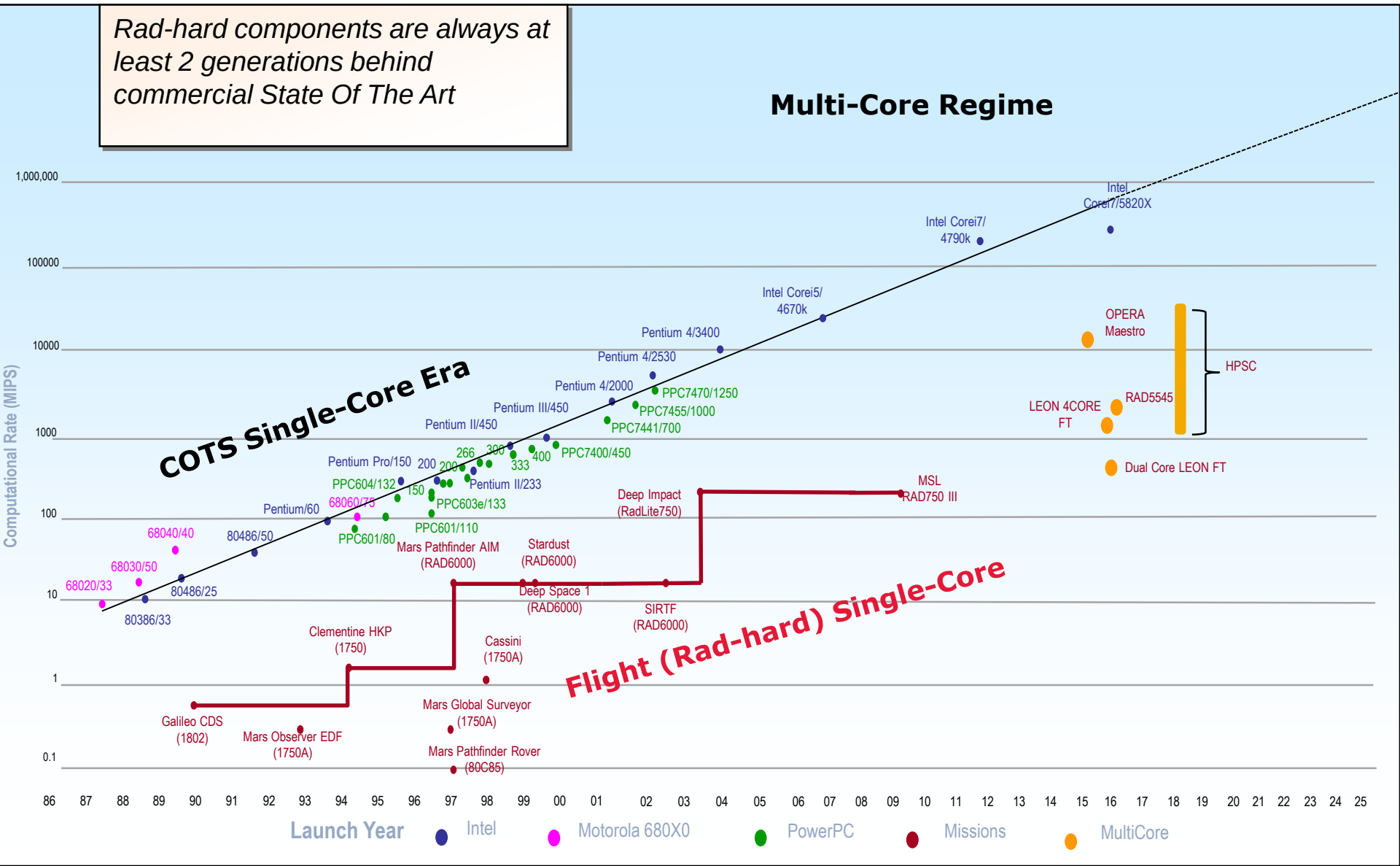
Raphael R. Some
Chief Technologist: Autonomous Systems Division
Caltech JPL
NASA Technical Authority: High Performance Spaceflight Computing
rsome@jpl.nasa.gov

Jet Propulsion Laboratory
California Institute of Technology

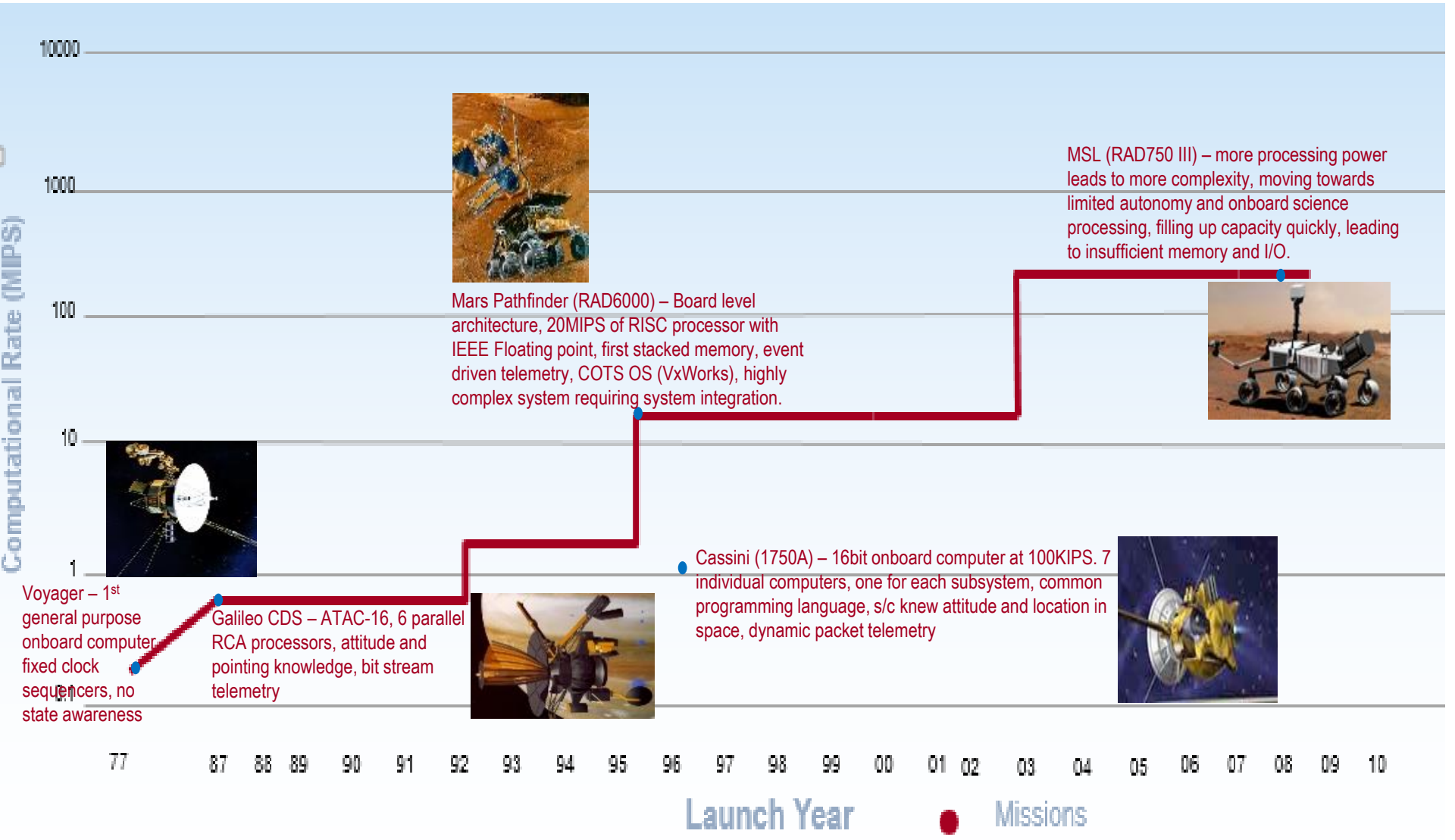
The work described in this publication was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract from the National Aeronautics and Space Administration under the STMD GCD Program. Government sponsorship acknowledged.

Space Flight Avionics and Microcomputer Processor History

Rad-hard components are always at least 2 generations behind commercial State Of The Art



Space Flight Avionics
Radiation Hardened Processors in Space



2012 Use Case Study

Human Spaceflight (HEOMD) Use Cases

1. Cloud Services
2. Advanced Vehicle Health Management
3. Crew Knowledge Augmentation Systems
4. Improved Displays and Controls
5. Augmented Reality for Recognition and Cataloging
6. Tele-Presence
7. Autonomous & Tele-Robotic Construction
8. Automated Guidance, Navigation, and Control (GNC)
9. Human Movement Assist

Science Mission (SMD) Use Cases

1. Extreme Terrain Landing*
2. Proximity Operations / Formation Flying*
3. Fast Traverse
4. New Surface Mobility Methods
5. Imaging Spectrometers*
6. Radar*
7. Low Latency Products for Disaster Response
8. Space Weather
9. Science Event Detection and Response*
10. Immersive Environments for Science Ops / Outreach

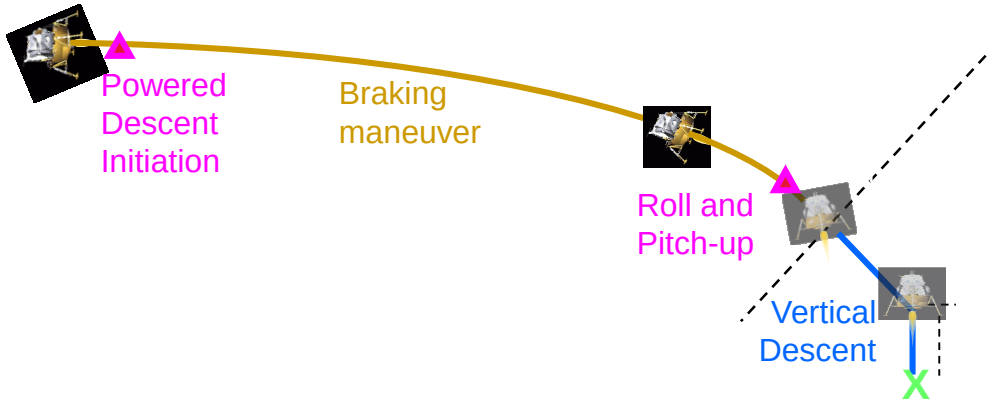
High value and mission critical applications identified by NASA scientists and engineers

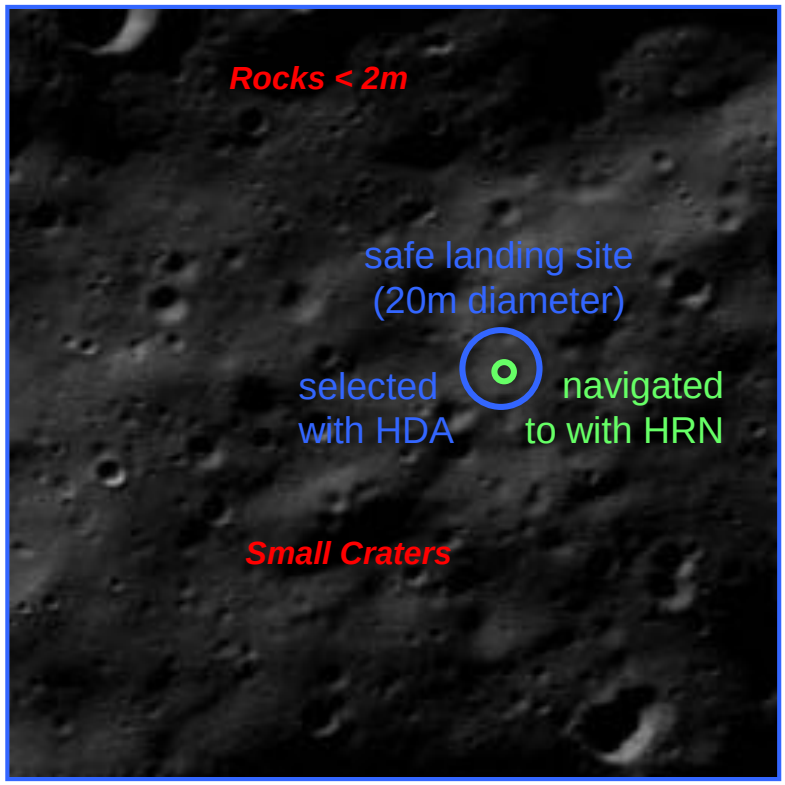
ALHAT
Extreme Terrain Pinpoint Landing

TRN = Terrain Relative Navigation

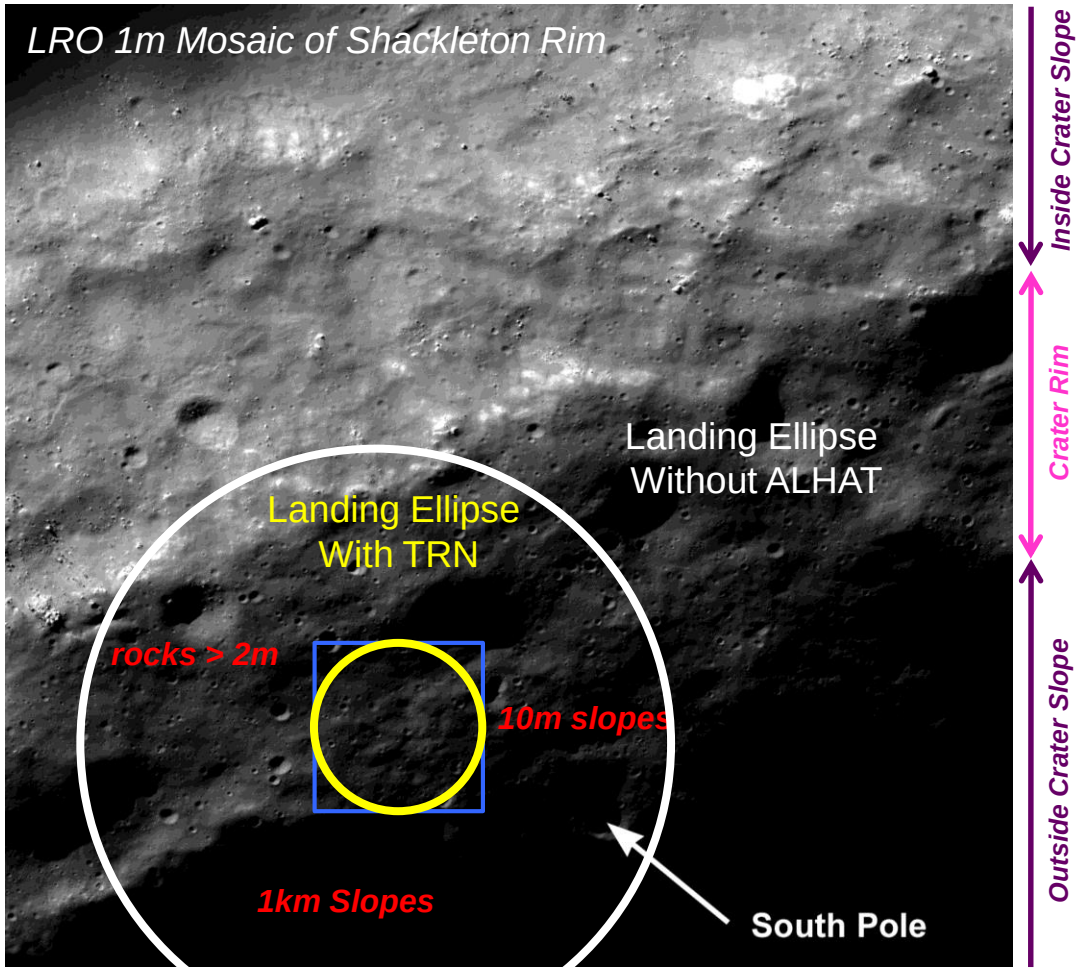
HDA = Hazard Detection and Avoidance

HRN = Hazard Relative Navigation



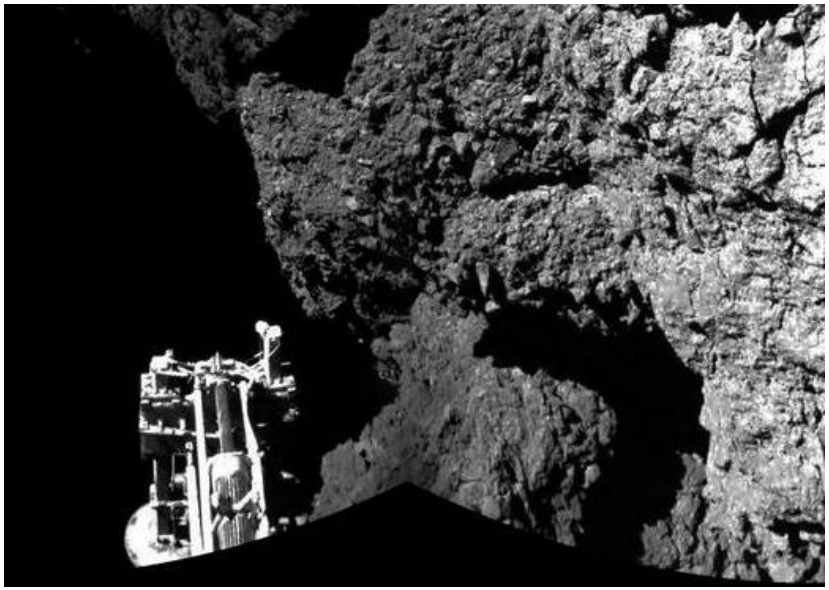


Hazard Map Area



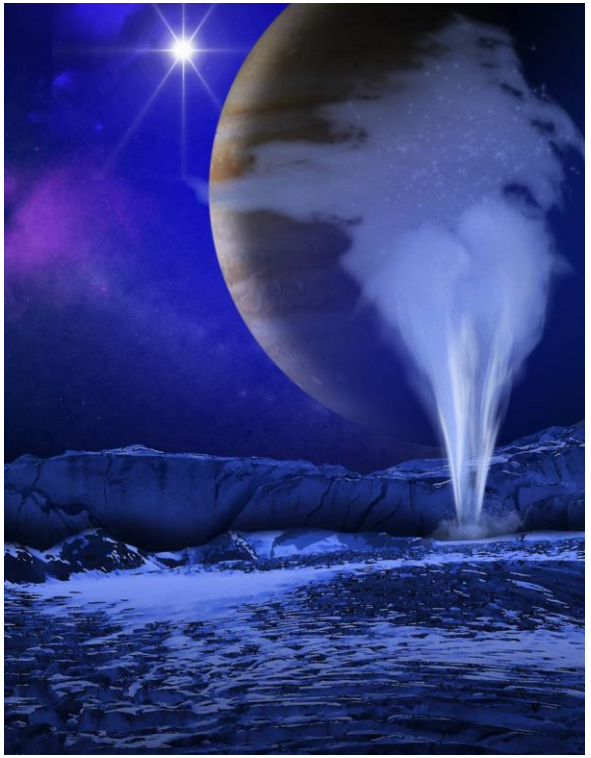
Proximity Operations

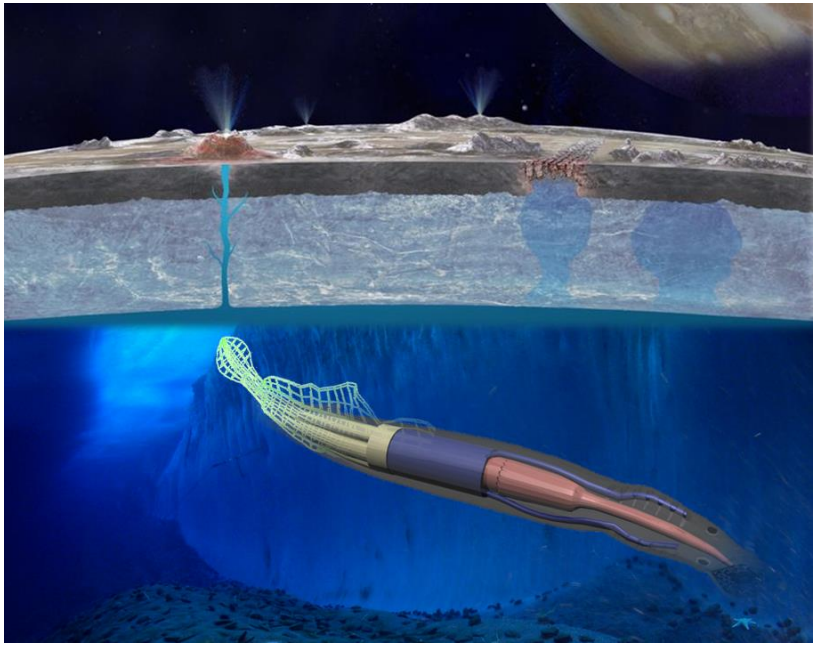
- Philae lander
- Concept mission Comet Hitchhiker



Autonomous Science Detection

Using new kinds of rovers (aquatic, aerial) to explore new environments and terrain.
Ability to autonomously and opportunistically observe phenomena of science interest.



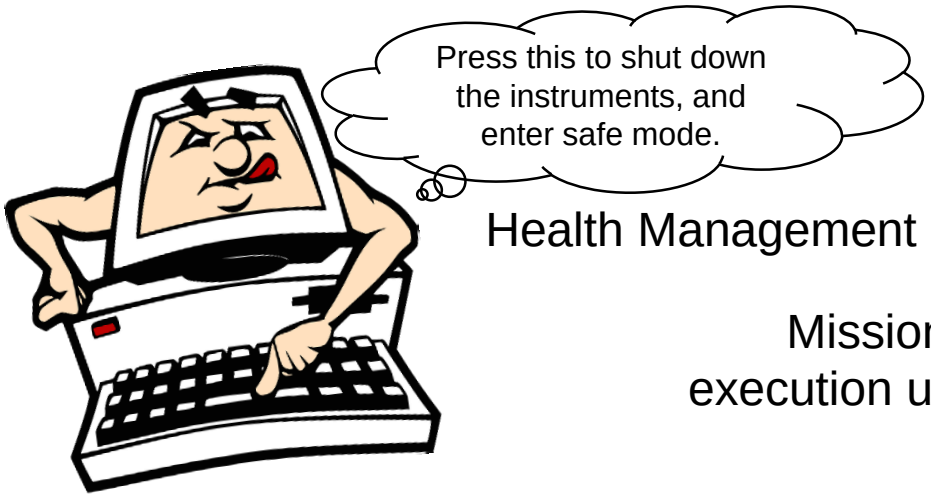


Computation Category	Mission Need	Objective of Computation	Flight Architecture Attribute	Processor Type and Requirements
Vision-based Algorithms with Real-Time Requirements	• Terrain Relative Navigation (TRN) • Hazard Avoidance • Entry, Descent & Landing (EDL) • Pinpoint Landing	• Conduct safe proximity operations around primitive bodies • Land safely and accurately • Achieve robust results within available timeframe as input to control decisions	• Severe fault tolerance and real-time requirements • Fail-operational • High peak power needs	• **Hard real time / mission critical** • **Continuous digital signal processing (DSP) + sequential control processing (fault protection)** • **High I/O rate** • **Irregular memory use** • **General-purpose (GP) processor (10's – 100's GFLOPS) + high I/O rate, augmented by co-processor(s)**
Model-Based Reasoning Techniques for Autonomy	• Mission planning, scheduling & resource management • Fault management in uncertain environments	• Contingency planning to mitigate execution failures • Detect, diagnose and recover from faults	• High computational complexity • Graceful degradation • Memory usage (data movement) impacts energy management	• **Soft real time / critical** • **Heuristic search, data base operations, Bayesian inference** • **Extreme intensive & irregular memory use (multi-GB/s)** • **> 1GOPS GP processor arrays with low latency interconnect**
High Rate Instrument Data Processing	High resolution sensors, e.g., SAR, Hyper-spectral	• Downlink images and products rather than raw data • Opportunistic science	• Distributed, dedicated processors at sensors • Less stringent fault tolerance	• **Soft real time** • **DSP/Vector processing with 10-100's GOPS (high data flow)** • **GP array (10-100's GFLOPS) required for feature ID / triage**

HPC and Autonomy for Robotic Science and Exploration

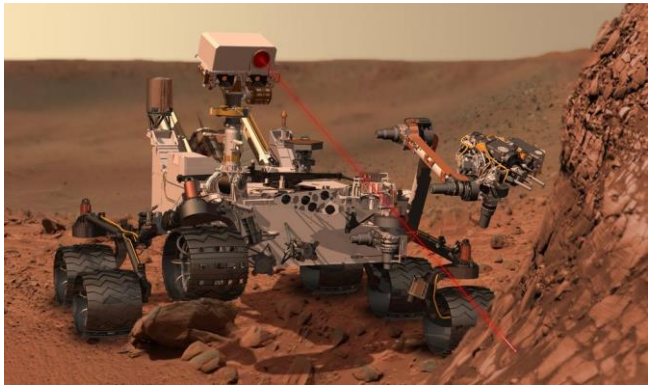
- Hierarchy of autonomy
  - First take care of yourself (*state awareness, fault handling*)
  - Next perform defined mission (*mission planning and execution* under changing conditions)
  - Third determine what to do in order to perform extended exploration and science, i.e., beyond explicitly specified (*goal driven mission planning, opportunistic science*) with available resources (*system capability knowledge)* in the changing environment (*situational awareness*)
  - Fourth cooperate with other robots (*constellation, team, swarm operations*)
  - Perform science data processing onboard, send back new knowledge (*autonomous science*)

Mission Visions - Robotics

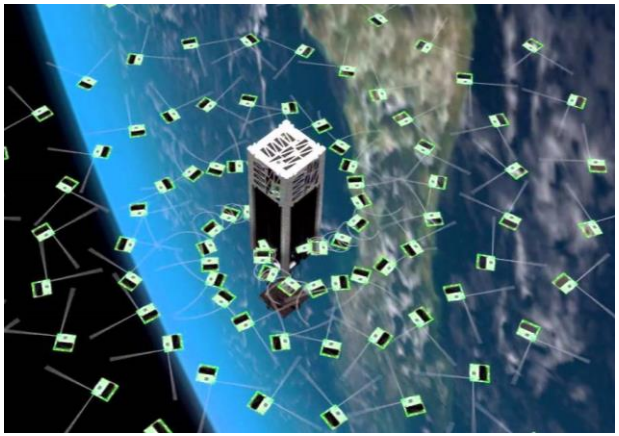


Health Management

Mission planning and execution under changing circumstances



Goal driven behavior



Spacecraft swarms

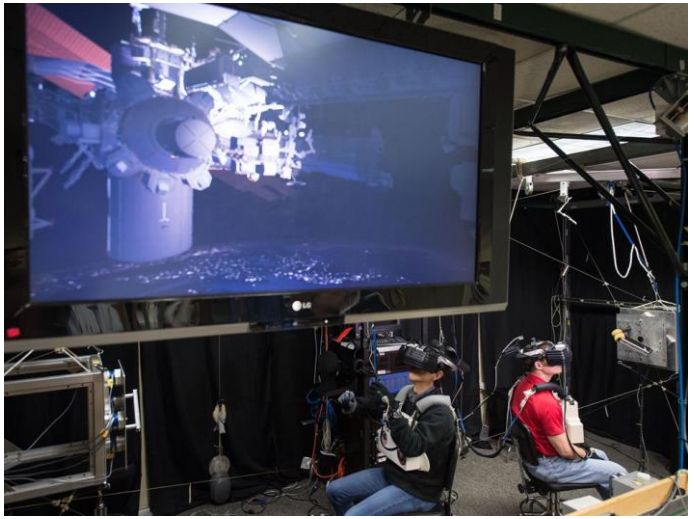
Onboard image processing

HPC and Autonomy for Crewed Missions

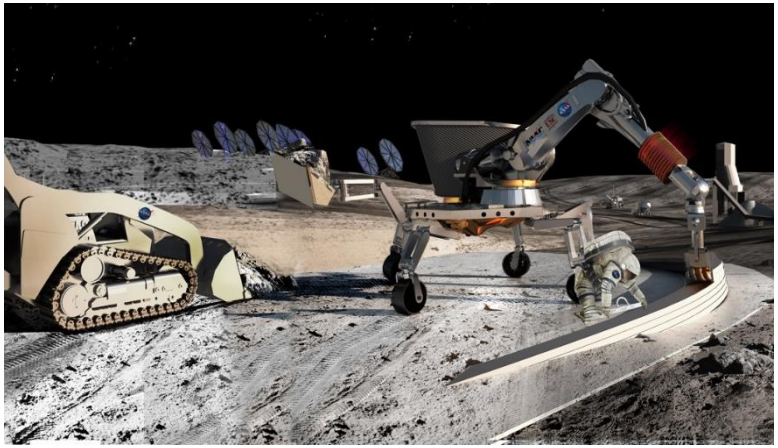
- Take care of vehicle and crew
  - Astronaut Assist
  - System health monitoring and diagnostics
  - Fault Handling
  - Maintenance and Repair
- Make it seem like they're at home
  - Virtual reality
  - Internet in space and delay tolerant communication
  - Augmented Reality
  - Visualizations of missions, remote crew & robotics operations
- Work independently for the human crew with minimal direction
  - Repairs, building habitats, or remote stations…
    - Teleoperation and virtual presence
- Work with the crew in a mixed team, understanding what to do with minimal direction and with maximal safely of humans
  - Medical lab, surgeries

Visions: Crewed missions

Autonomous repair



Virtual Reality, Augmented Reality



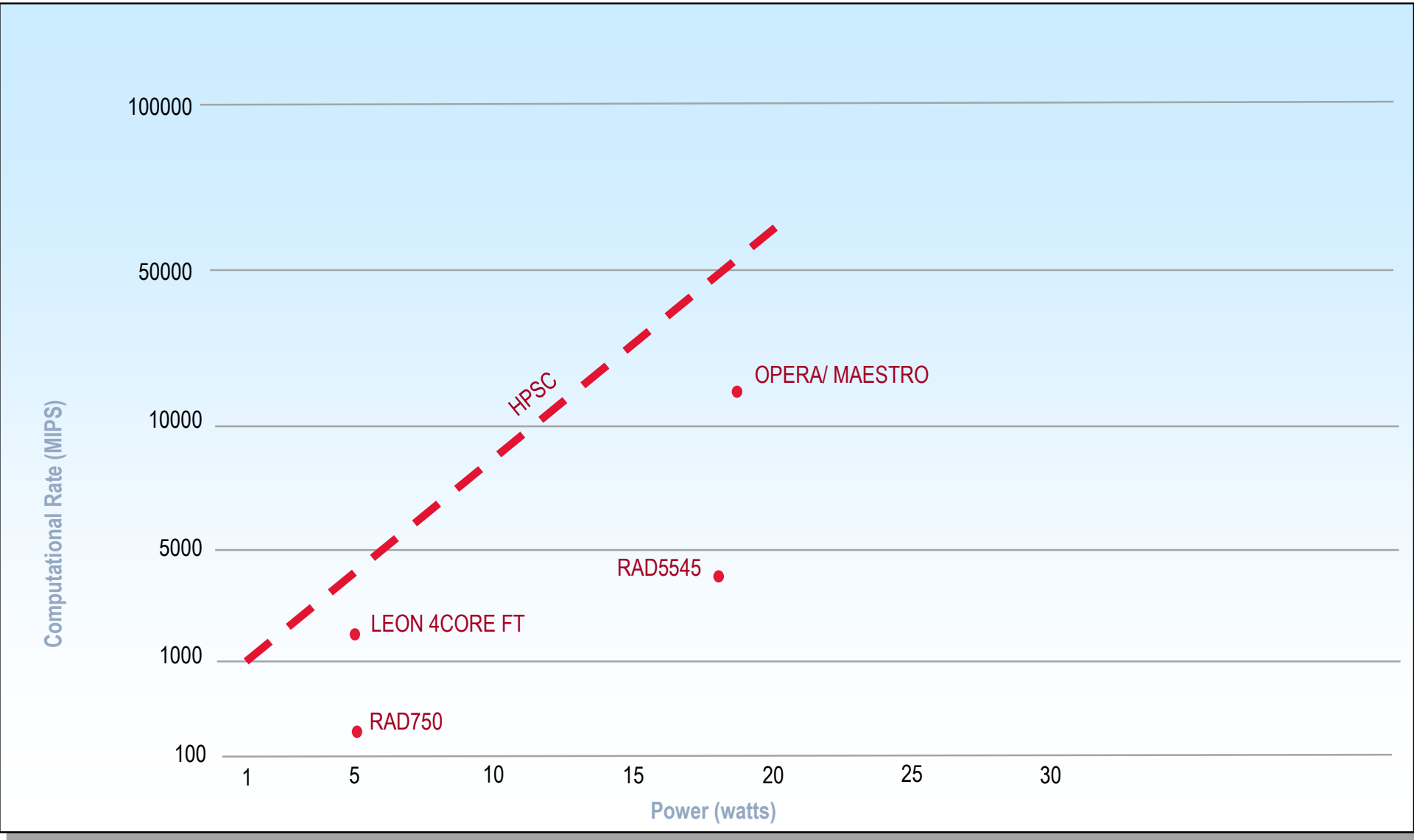
Work independently with minimal direction

Interact with humans on a task

What will we need to Achieve These Capabilities?

- Tens to Hundreds of GOPS of throughput
  - Specialized custom co-processors in a
  - Heterogeneous Computing Environment
- Extremely high reliability
  - Hardware
  - Software
  - System operation
- Ability to withstand faults and damage without compromising delivered service
- Ability to gracefully and intelligently degrade in performance while maintaining safety and high priority services
- Tens to Hundreds of Gb/s I/O data rates
- Tens to Hundreds of GB/s memory data rates
- Tens of TB of memory capacity
- At extremely low SWaP-C

Multicore Rad Hard Processors
Performance:Power

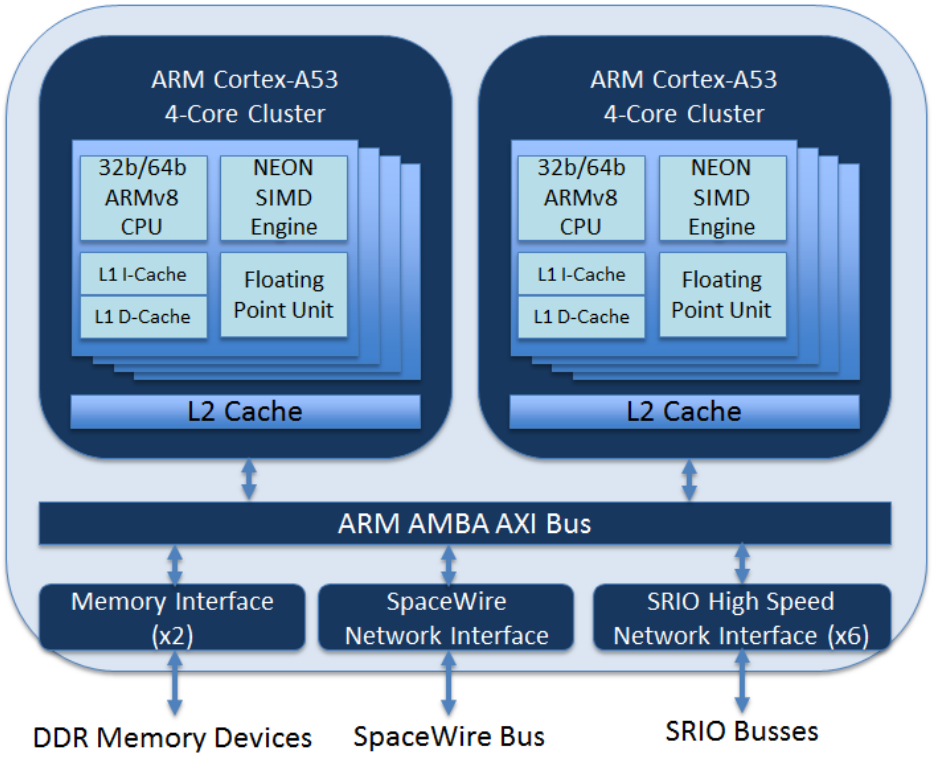


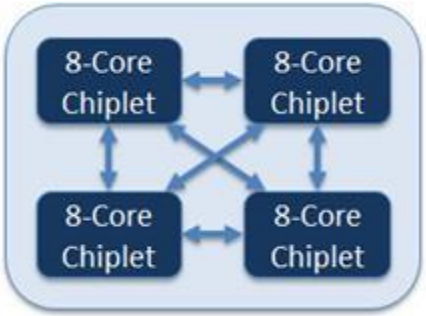
Current Multicore Processors

- Leon 4 & variants, RAD 5545 “System on a Chip” Architectures
- Self contained
  - Complete system, but limited extensibility
- 4 processing cores + I/O + memory interface
- Limited power management
- Limited fault tolerance strategies
- Limited resource utilization strategies
  - Binding of “subsystem” to processing core
    - CDH, GnC, Comm, Instrument processing & control
  - Classical SMP
    - Allocation of processor core to next task or thread
  - Other strategies, e.g. AMP, possible, but limited benefit vs complexity
- Bottlenecks can be a significant issue depending on application
  - Especially memory
- Straightforward programming with standard OS, compilers, debuggers

HPSC Reference Architecture

8-Core Extensible Chiplet





- A53 clusters for high bandwidth processing provide ~15 GOPS
- Typical device power is ~5-7 Watts (depending on memory and I/O utilization)
- On chip AMBA interconnect,
- 2 72-bit DDR3/4 memory interfaces
- 6 Serial RapidIO (SRIO) busses (10Gbaud each) to interconnect other chiplets, and high bandwidth instruments and subsystems
- 6 XAUI port (10Gbaud each)
- Misc I/O: NVM, SRAM, GPIO, Boot ROM
- Power Management – unused cores can be dynamically de-powered or put to sleep
- Multiple levels of fault tolerance – hardware and software implemented – some mandatory, some optional

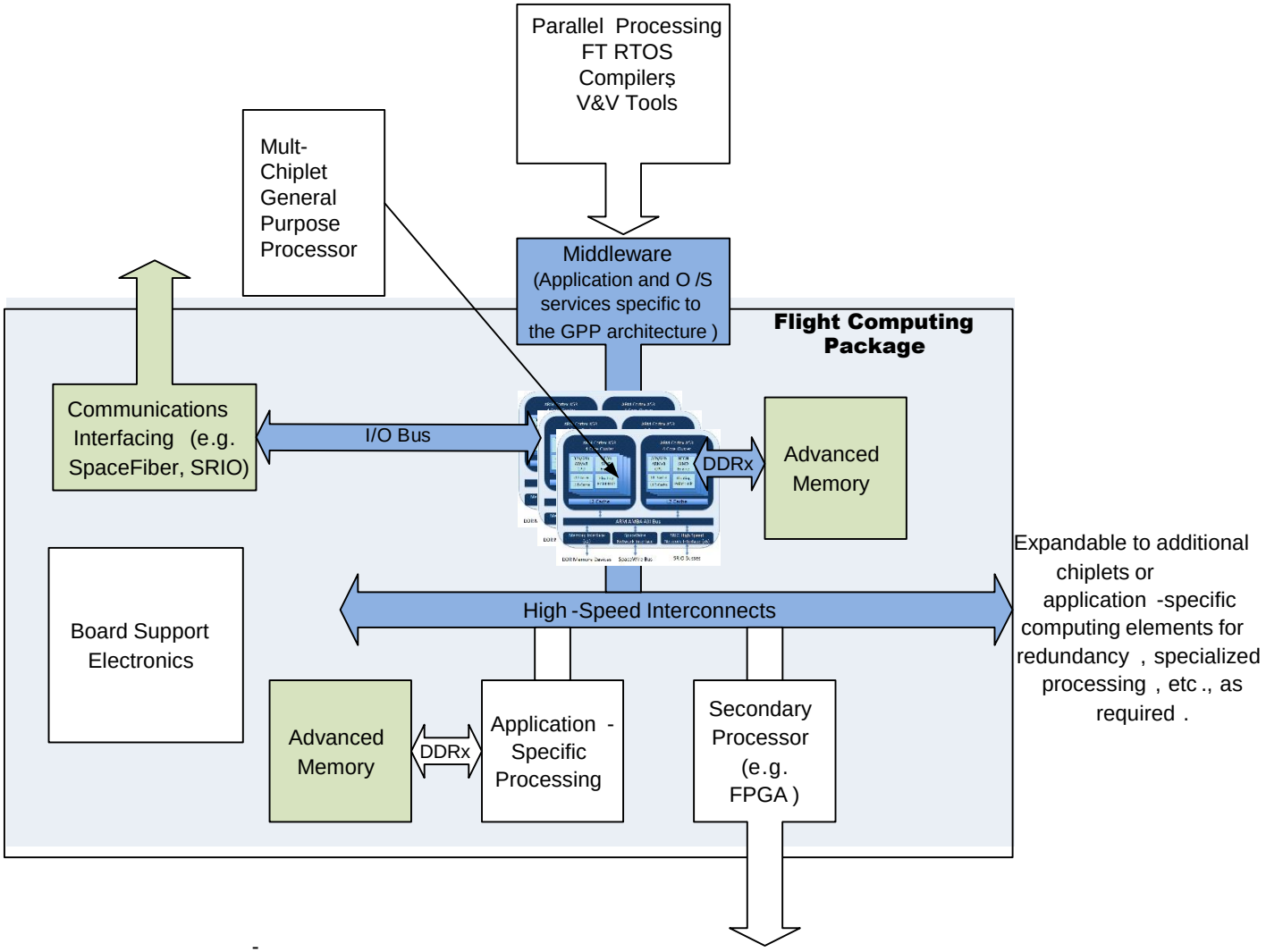
The Chiplet Concept

- System in Package (SIP) vs System on a Chip
  - Build complex systems from small, reusable modules, aka Chiplets
  - Flexibility/Scalability
    - Multiple Chiplets in arbitrary topology
    - Mix & match Chiplet technologies/generations
    - Multiple modes/levels of fault tolerance, power, dynamically manageable
    - Single Chip, 2.5D, 3D packaging (& chiplet configurations)
  - Extensibility
    - Coprocessors: PIM, Neuromorphic, Robotic, DSP via SRIO
    - FPGA via XAUI/SRIO
  - Evolvability
    - Low cost, rapid evolution of:
      - Chiplets
      - Chiplet-based SIP Computers
  - Affordability
    - Low cost, rapid development
      - Chiplet
      - Processor/computer package/board

The Chiplet Concept

- But at the cost of complexity
  - Software needs to handle
    - Multiple widely varying hardware configurations and capabilities
    - Parallel processing (not just multithreading)
    - Dynamically varying hardware resources
    - Dynamically varying software loads with different optimization strategies
  - Software development challenge
    - Rapid (10M LOC in 2-3 years)
    - Highly reliable
    - 100% V&V coverage
  - Spacecraft System Level
    - Distributed computing as well as centralized
    - Fault tolerance
    - Code migration

HPSC Ecological Elements



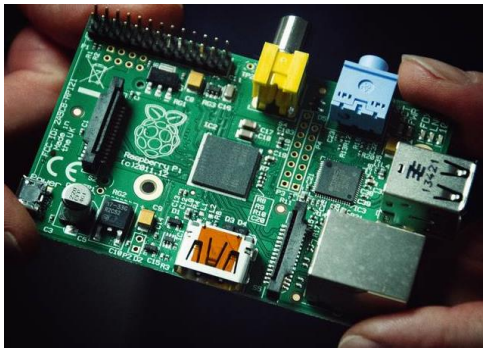
The overall High Performance Spaceflight Computing (HPSC) architecture is an “ecology” formed by the processor and supporting hardware and software elements to make a modern, scalable, rad hard computing environment.

Challenges

- Hardware:
  - Cost!
  - Heterogeneous configurations – tailored to application
  - Keep up with COTS capabilities and tools
  - Robustness, life time, radiation hardness…. in a shrinking geometry
  - Can we build a rapidly evolving, easy to use, plugnplay tool kit?
- Software:
  - 10s of Millions of LOC in 2-3 years guaranteed correct and dynamically V&V'd with 100% coverage
  - Easy to use development system
  - Highly complex parallel codes for science and autonomy
  - At an affordable price!

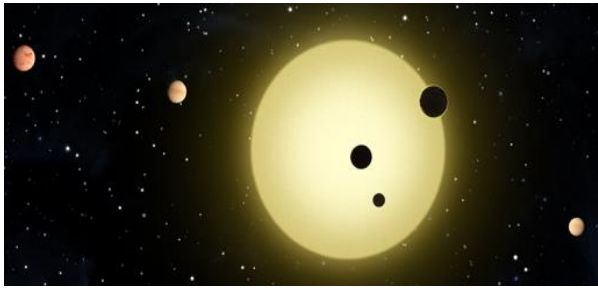
Do we want to show a conceptual architecture – I think so??

Final Thoughts



COTS

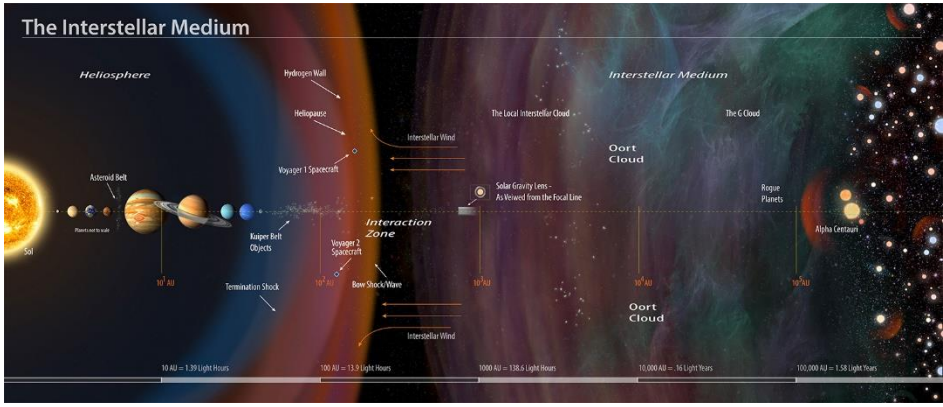
+ $ +



COMPLEXITY

+

=



Jet Propulsion Laboratory
California Institute of Technology